

Article

CAC: A Learning Context Recognition Model Based on AI for Handwritten Mathematical Symbols in e-Learning Systems

Sung-Bum Baek ¹, Jin-Gon Shon ² and Ji-Su Park ^{3,*}

¹ Department of e-Learning, Graduate School, Korea National Open University, Seoul 03087, Korea; juinha@knou.ac.kr

² Department of Computer Science, Korea National Open University, Seoul 03087, Korea; jgshon@knou.ac.kr

³ Department of Computer Science and Engineering, Jeonju University, Jeonju 55069, Korea

* Correspondence: jisupark@jj.ac.kr; Tel.: +82-63-220-2249

Abstract: The e-learning environment should support the handwriting of mathematical expressions and accurately recognize inputted handwritten mathematical expressions. To this end, expression-related information should be fully utilized in e-learning environments. However, pre-existing handwritten mathematical expression recognition models mainly utilize the shape of handwritten mathematical symbols, thus limiting the models from improving the recognition accuracy of a vaguely represented symbol. Therefore, in this paper, a context-aided correction (CAC) model is proposed that adjusts an output of handwritten mathematical symbol (HMS) recognition by additionally utilizing information related to the HMS in an e-learning system. The CAC model collects learning contextual data associated with the HMS and converts them into learning contextual information. Next, contextual information is recognized through artificial intelligence to adjust the recognition output of the HMS. Finally, the CAC model is trained and tested using a dataset similar to that of a real learning situation. The experiment results show that the recognition accuracy of handwritten mathematical symbols is improved when using the CAC model.



Citation: Baek, S.-B.; Shon, J.-G.; Park, J.-S. CAC: A Learning Context Recognition Model Based on AI for Handwritten Mathematical Symbols in e-Learning Systems. *Mathematics* **2022**, *10*, 1277. <https://doi.org/10.3390/math10081277>

Academic Editor: Paolo Crippa

Received: 26 January 2022

Accepted: 9 April 2022

Published: 12 April 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: handwritten mathematical symbol recognition; learning context; contextual data; contextual information

MSC: 68T10; 97U50; 97U70

1. Introduction

Numerous symbols with an explicit meaning are used in various mathematical expressions. However, symbols in handwritten mathematical expressions are often vaguely expressed for various reasons, including the handwriting style of the individual writer and the characteristics of the input tool. Therefore, even in datasets widely applied in handwritten mathematical expression recognition research, many vaguely expressed symbols exist.

Therefore, to recognize a handwritten mathematical symbol (HMS) more accurately, it is necessary to consider not only the shape of the HMS but also the data surrounding the HMS, that is, the contextual data. The contextual data of an HMS can be broadly divided into contextual data inside the expression and contextual data outside the expression. Figure 1 shows two examples of HMS recognition errors. Among them, Figure 1a shows a case in which the contextual data inside the expressions must be considered. If referring to the other symbols in the expression, the incorrectly recognized “v” can be corrected as “a”. By contrast, Figure 1b shows a case in which the contextual data outside the expression must be considered. Here, when referring to the symbols used in the first two entered expressions, the incorrectly recognized “u” can be modified as “a”.

Human-related data are ambiguous and diverse; therefore, it is necessary to utilize contextual data to process them accurately. Accordingly, studies using contextual data have been conducted to accurately recognize complex data, such as human behavior and

living environments [1,2]. However, no studies have been conducted on the recognition of an HMS that sufficiently consider contextual data in e-learning environments. This paper proposes a use of contextual data outside the expression, which are obtained from an e-learning system.

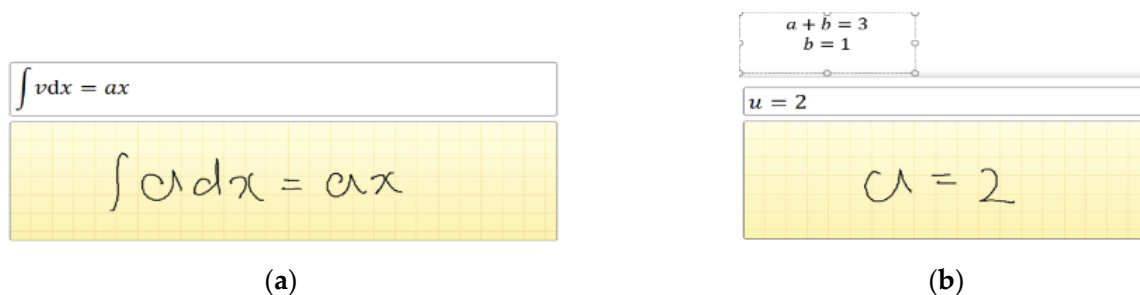


Figure 1. Samples of HMS misrecognitions. (a) Missing contextual data inside the expression. (b) Missing contextual data outside the expression.

Throughout this paper, learning context (LC) refers to the environment that influences the learning, such as the learning contents and learning situations. Accordingly, the data in the e-learning system, which are related to the data generated by the learner during learning, are defined as learning contextual data (LC data). In addition, information converted to allow LC data to be used directly for functions including automatic computer recognition is defined as learning contextual information (LC information).

This paper describes a method for adjusting an output of HMS recognition (HMS output) by effectively using LC data. To this end, symbols in mathematical expressions extracted from the learning contents and system data regarding the input positions of these are used as LC data. In addition, LC information is generated using LC data so that it can be directly used to adjust the HMS output. By recognizing LC information through artificial intelligence and correcting the HMS output, the effect of using learning context is proven. The symbols and range of the learning contents used in the implementation and experiment were limited to specific units of middle school mathematics, and the LC data was randomly generated but configured similarly to an actual workbook.

2. Related Work

2.1. Handwritten Mathematical Expression Recognition

Handwritten mathematical expressions refer to expressions written by a user by hand with a pen or similar tool. In an e-learning environment, handwritten mathematical expressions are generally stored as digital data in the form of images and can be broadly divided into offline handwritten mathematical expressions and online handwritten mathematical expressions. Offline handwritten mathematical expressions contain only pixel data, such as general photographic images, whereas online handwritten mathematical expressions include stroke data obtained through a stylus or finger touch, that is, both coordinates of the points and temporal sequence data [3]. This paper aims to identify online handwritten mathematical symbols that can utilize LC data. Therefore, the handwritten mathematical expressions and symbols mentioned in this paper refer to online ones.

As shown in Figure 2, the recognition process for handwritten mathematical expressions is divided into symbol segmentation, symbol recognition, and structural analysis. Whereas symbol segmentation is the process of grouping one or more strokes in a handwritten mathematical expression and dividing them into individual symbol images, symbol recognition is the process of recognizing each symbol image and converting the images into text-format data. A structural analysis is the process of identifying the spatial relations between symbols in consideration of their size and position [3–6].

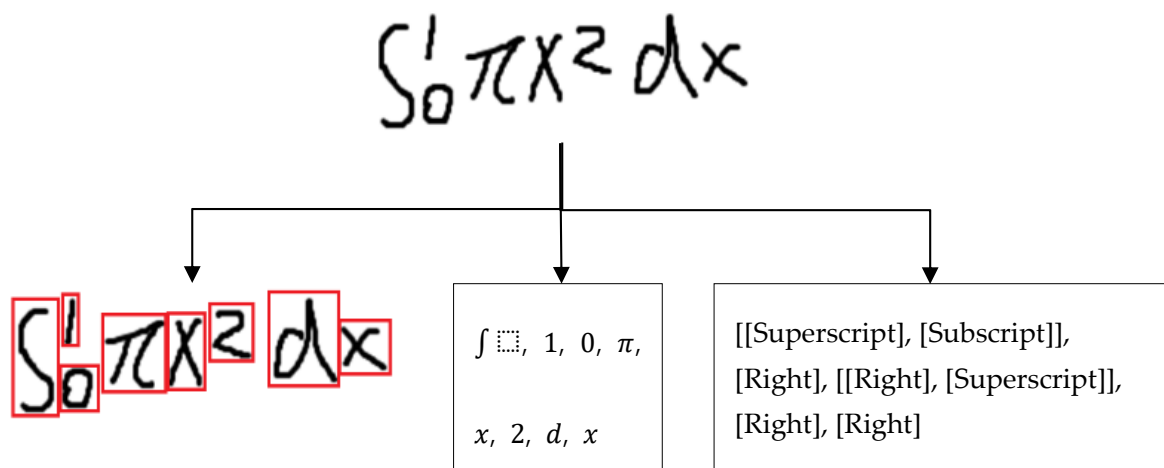


Figure 2. Recognition process of handwritten mathematical expressions.

In the initial studies on handwritten mathematical expression recognition, the recognition process shown in Figure 2 was sequentially carried out according to an order; however, there is a limitation in that the incorrect output of the previous process affects the following process, and the contextual data inside the expression are not considered [5]. Owing to these difficulties, although, many studies have attempted to apply all recognition processes simultaneously as follows, a perfect level has yet to be reached.

- Geometric convex hull constraint, A-star completion estimate, book-keeping [7]
- Simultaneous segmentation and recognition through hidden Markov model (HMM) approach [8]
- Simultaneous segmentation and recognition through probabilistic context-free grammar [9]
- Gaussian mixture model, bidirectional long short-term memory (BLSTM) and recurrent neural network (RNN), two-dimensional probabilistic context-free grammars [10]
- BLSTM, Cocke–Younger–Kasami algorithm (CYK) [11]

In particular, studies using two-dimensional probabilistic context-free grammar, HMM, and contextual information inside the expression have been conducted to solve the ambiguous symbol recognition problem. However, it has been difficult to obtain efficient recognition results because of symbols that have similar shape but different semantics, such as $\{1, |, ', \text{comma}\}$, $\{P, p\}$, $\{S, s\}$, $\{C, c\}$, $\{X, x, \times\}$, $\{V, v\}$ and $\{o, 0\}$ [12,13].

2.2. Pre-Existing Handwritten Mathematical Expression Recognition Models

The Competition on Recognition of Handwritten Mathematical Expressions (CROHME) is held to encourage handwritten mathematical expression recognition research. It provides available data and evaluates the system performance using the same platform and testing data. Numerous research teams have participated in six competitions from 2011 to 2019. Three tasks were applied at 2019 CROHME: online handwritten mathematical expression recognition (Task 1), offline handwritten mathematical expression recognition (Task 2), and the detection of expressions in document pages (Task 3). Among them, the subtasks of recognizing isolated symbols, Tasks 1a and 2a, and parsing expressions from the provided symbols, Tasks 1b and 2b, were added for Tasks 1 and 2, respectively. For these tasks, CROHME provided 12,178 expression data, 214,358 symbol data, 12,126 structure data, and 38,280 expression detection data [14]. The experiment conducted in this paper used the symbol dataset provided for the online single-symbol recognition task (Task 1a) at 2019 CROHME.

The online handwritten mathematical expression recognition task (Task 1) at 2019 CROHME involved eight research teams, as shown in Table 1 [14]. The team that obtained the highest recognition rate was USTC-iFLYTEK (USTC-NELSLIP and iFLYTEK Research), who achieved an accuracy of 80.73% in the simultaneous recognition of expression struc-

tures and symbols, whereas the recognition accuracy when considering only the expression structure and ignoring the symbol recognition result was 91.49%. The fact that the accuracy of the symbol recognition barely exceeded 80% means that the symbol was incorrectly recognized once every time five expressions were input in a real learning environment, which is a level at which learners can still feel uncomfortable.

Table 1. Online handwritten mathematical expressions recognition results.

No	Team	Model (Based Method)	Recognition Data	Accuracy	
				Structure + Symbol Labels	Structure
1	USTC-iFLYTEK	TAP (RNN ¹) [15]	Online data	80.73%	91.49%
2	Samsung R&D 1	PCFG (RNN ¹ , PCFG ²) [11]	Online data	79.82%	89.32%
3	MyScript	MyScript Math recognizer (BLSTM ³ , LSTM ⁴) [14]	Online data	79.15%	90.66%
4	Sun Yat-Sen U.	MyScript Interactive Ink [4]	Online data extracted from images	77.40%	88.82%
5	Samsung R&D 2	Text/shape classifier (SVM ⁵) [16]	Online data	65.97%	82.82%
6	PAL-v2	PAL-v2 (LSTM ⁴) [17]	Images converted from online data	62.55%	79.15%
7	MathType	MathType (LSTM ⁴) [14]	Images converted from online data	60.13%	79.15%
8	TUAT	body box (LSTM ⁴ , PCFG ³ , SVM ⁵) [18]	Online data and offline data (converted from online data)	39.95%	58.22%

¹ Recurrent Neural Network, ² Probabilistic Context-free Grammar, ³ Bidirectional Long Short-Term Memory, ⁴ Long Short-Term Memory, ⁵ Support Vector Machine.

Further analysis of the misrecognized results of all teams suggests that errors in the structure recognition commonly lead to errors in symbol recognition [14]. In addition, it can be interpreted that there are many handwritten mathematical expressions in which the information on the structure did not help in recognizing the symbols, even when the structure was properly recognized. Taking the results of the USTC-iFLYTEK team as an example, in 8.51% of all data, the structure was incorrectly recognized, and many structural errors caused errors in symbol recognition. In addition, in 10.76% of all data, although a correct structure recognition was achieved, an error occurred in the symbol recognition. In most cases, information on the structure recognition outputs was not utilized or was insufficient for recognition of an ambiguous symbol.

An RNN is a representative artificial neural network used to recognize handwritten mathematical expressions. Because RNNs are suitable for recognizing sequential data of variable lengths, they have been used in various studies, including a document summary and email traffic modeling [19,20]. Because the length of online handwritten mathematical expression data is not fixed, RNNs are typically used for the data recognition [15]. In particular, long short-term memory (LSTM), an improved RNN model, adds an input gate, a forget gate, and an output gate to the memory cells of the hidden layer. They remove unnecessary memories from the cell state or add specific information required to it concerning the inputs and the hidden states. All information can be linked to other information with relatively large time intervals through the cell state [21,22]. As shown in Table 1, many of the teams participating in the online handwritten mathematical expression recognition task at 2019 CROHME used RNNs or LSTM.

3. LC Data

3.1. Composition of Learning Contents

As shown in Figure 3, the learning contents stored in the e-learning system described in this paper are composed of four types of learning parts: the learning topics, questions, solving processes, and correct answers. The k th learning topic S^k contains expressions $\sigma_1^k, \sigma_2^k, \dots$. The questions related to the learning topic S^k are $Q_1^k, \dots, Q_{N_k}^k$. The l th question Q_l^k ($1 \leq l \leq N_k$) contains expressions $\rho_{(l)1}^k, \rho_{(l)2}^k, \dots$. In addition, W_l^k , which is the solving process related to the question Q_l^k , contains expressions $\omega_{(l)1}^k, \omega_{(l)2}^k, \dots$. Here, A_l^k , which is the correct answer related to the question Q_l^k , contains expressions $\alpha_{(l)1}^k, \alpha_{(l)2}^k, \dots$.

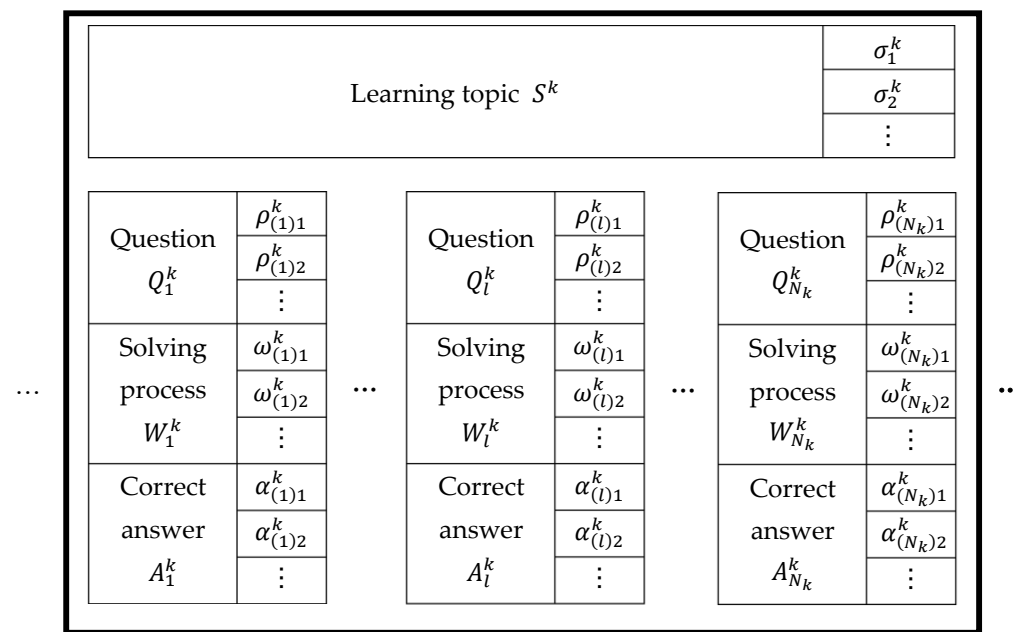


Figure 3. Composition of learning contents stored in e-learning systems.

The universal set U of learning parts in the e-learning system and subsets of U according to types of learning parts are defined as follows.

$$U = \{P \mid P \text{ is a learning part in the e-learning system}\} \tag{1}$$

$$U_1 = \{P \in U \mid P \text{ is a learning topic}\} = \{S^1, S^2, \dots\} \tag{2}$$

$$U_2 = \{P \in U \mid P \text{ is a question}\} = \{Q_1^1, Q_2^1, \dots, Q_1^2, Q_2^2, \dots\} \tag{3}$$

$$U_3 = \{P \in U \mid P \text{ is a solving process}\} = \{W_1^1, W_2^1, \dots, W_1^2, W_2^2, \dots\} \tag{4}$$

$$U_4 = \{P \in U \mid P \text{ is a correct answer}\} = \{A_1^1, A_2^1, \dots, A_1^2, A_2^2, \dots\} \tag{5}$$

For a learning part $P \in U$, $L_1(P)$, $L_2(P)$, $L_3(P)$, and $L_4(P)$ are defined as learning part sets of learning topics, questions, solving processes, and correct answers related to P , respectively.

$$L_1(P) = \{P' \in U_1 \mid P' \text{ is a learning topic related to } P\} \tag{6}$$

$$L_2(P) = \{P' \in U_2 \mid P' \text{ is a question related to } P\} \tag{7}$$

$$L_3(P) = \{P' \in U_3 \mid P' \text{ is a solving process related to } P\} \tag{8}$$

$$L_4(P) = \{P' \in U_4 \mid P' \text{ is a correct answer related to } P\} \tag{9}$$

For example, $L_1(Q_1^1) = \{S^1\}$, $L_2(S^1) = \{Q_{11}^1, Q_{12}^1, \dots, Q_{1N_1}^1\}$, $L_3(Q_1^1) = \{W_1^1\}$, and $L_4(S^1) = \{A_1^1, A_2^1, \dots, A_{N_1}^1\}$.

For a learning part P , $F(P)$ is defined as a set of all expressions included in P .

$$F(P) = \{\varphi \mid \varphi \text{ is an expression in } P\} \tag{10}$$

For example, $F(W_1^1) = \{\omega_{(1)1}^1, \omega_{(1)2}^1, \dots\}$.

3.2. Extracted Symbol and Input Position

The expressions in the learning contents of each learning part contain symbols. An extracted symbol is defined as a symbol extracted from the expressions in the learning contents. Table 2 lists an example of extracted symbols. Because the learner inputs mathematical expressions based on these symbols, it is necessary to use the extracted symbols as LC data for correcting the outputs of the ambiguously expressed symbols in the HMS recognition algorithm.

Table 2. Sample symbols extracted from learning contents.

Learning Part	Learning Contents	Expressions	Extracted Symbols
Learning topic	<Linear Inequality> When the terms on the right side of the inequality are transposed to the left side, the inequality that appears in either (Linear Expression) < 0, (Linear Expression) > 0, (Linear Expression) ≤ 0, and (Linear Expression) ≥ 0 is called the linear inequality.	(Linear Expression) < 0	<, 0
		(Linear Expression) > 0	>, 0
		(Linear Expression) ≤ 0	≤, 0
		(Linear Expression) ≥ 0	≥, 0
Question	Find the range of values of the constant a when the root of equation $x - 2 = \frac{x+a}{3}$ is not greater than 1.	a	a
		$x - 2 = \frac{x+a}{3}$	$x, a, -, =, +, -, 2, 3$
		1	1
Solving process	$3x - 6 = x + a$ $2x = a + 6$ $x = \frac{a+6}{2}$ Since $x \leq 1$, $\frac{a+6}{2} \leq 1$ $a + 6 \leq 2$	$3x - 6 = x + a$	$x, a, -, =, +, 3, 6$
		$2x = a + 6$	$x, a, =, +, 2, 6$
		$x = \frac{a+6}{2}$	$x, a, =, +, -, 6, 2$
		$x \leq 1$	$x, \leq, 1$
		$\frac{a+6}{2} \leq 1$	$a, +, -, \leq, 6, 2, 1$
		$a + 6 \leq 2$	$a, +, \leq, 6, 2$
Correct answer	$a \leq -4$	$a \leq -4$	$a, \leq, -, 4$

For an expression φ , $S(\varphi)$ is defined as a set of all extracted symbols in φ .

$$S(\varphi) = \{s \mid s \text{ is an extracted symbol in } \varphi\} \tag{11}$$

For example, $S(a \leq -4) = \{a, \leq, -, 4\}$.

For a symbol s and a learning part set X , $\bar{P}(s, X)$ is defined as a set of all learning parts containing s within X .

$$\bar{P}(s, X) = \{P \in X \mid s \in \bigcup_{\varphi_k \in F(P)} S(\varphi_k)\} \tag{12}$$

Therefore, $\bar{P}(s, U_1)$, $\bar{P}(s, U_2)$, $\bar{P}(s, U_3)$, and $\bar{P}(s, U_4)$ are sets of learning topics, questions, solving processes, and correct answers that include an expression containing symbol s , respectively.

The input position of the expression is also used as LC data in this paper. As shown in the example in Figure 4, there are two types of places where a learner enters an expression during learning: solving processes and answers. The symbols that learners primarily use at each position are different. In addition, even if the same symbol is used for each position, the meaning may be interpreted differently.

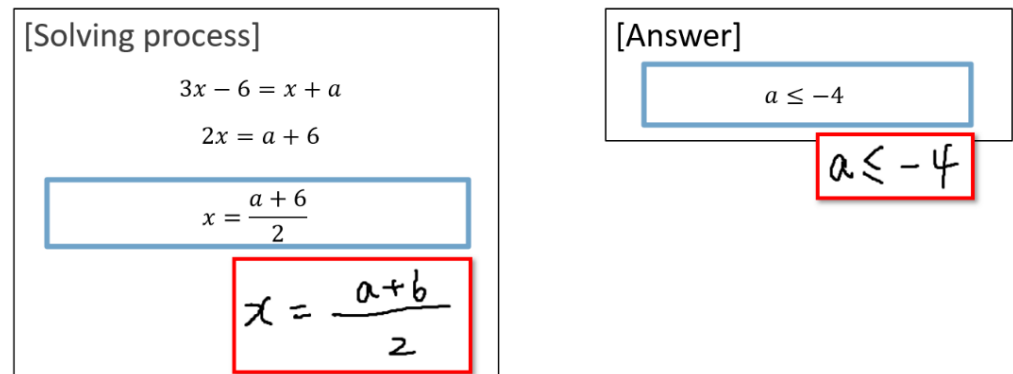


Figure 4. Input position of expression.

3.3. LC Data from e-Learning System

In this paper, it is assumed that learners try to write similar solving processes and answers to model-solving processes and correct answers, respectively, as much as possible with reference to contents in learning topics and questions, and that the following data can be obtained as LC data along with HMS x from an e-learning system.

- $P_1(x)$ is the learning topic that the learner is studying when x is input.
- $P_2(x)$ is the question that the learner is solving when x is input.
- $P_3(x)$ is the solving process of the question that the learner is solving when x is input.
- $P_4(x)$ is the correct answer of the question that the learner is solving when x is input.
- The input position $i(x)$ is the value indicating which type of learning part x is input in.

$$i(x) = \begin{cases} 0, & x \text{ is input in a solving process.} \\ 1, & x \text{ is input in an answer.} \end{cases} \quad (13)$$

We defined symbol list D , which is an ordered list of all symbols used as an output of the HMS recognition. The symbol list size n_D is the total number of symbols in symbol list D , and $d_i \in D$ ($1 \leq i \leq n_D$) represents the i th symbol of symbol list D , where i is the index of symbol d_i .

The HMS information used in this paper is a row vector expressed as $[p_1 \ p_2 \ \dots \ p_{n_D}]$. Each element p_i ($1 \leq i \leq n_D$) of the HMS information is the probability that the interpretation of symbol $d_i \in D$ is the correct one. Given HMS x , two vectors for the HMS information are used: the HMS output $y_o(x) = [p_1^o \ p_2^o \ \dots \ p_{n_D}^o]$, which is the recognition output of HMS x , and the context-applied output $y_r(x) = [p_1^r \ p_2^r \ \dots \ p_{n_D}^r]$, which is the adjusted output of the HMS output $y_o(x)$ that reflects the LC information.

The definitions of all symbols and functions in Section 3 are summarized in Appendix A.

4. CAC Model

4.1. Composition of CAC Model

In this paper, a context-aided correction (CAC) model is designed as a method for correcting the HMS output using the learning context. It consists of three parts: an LC data collection module, LC information generation module, and HMS output correction module. The composition and function of each module are shown in Figure 5.

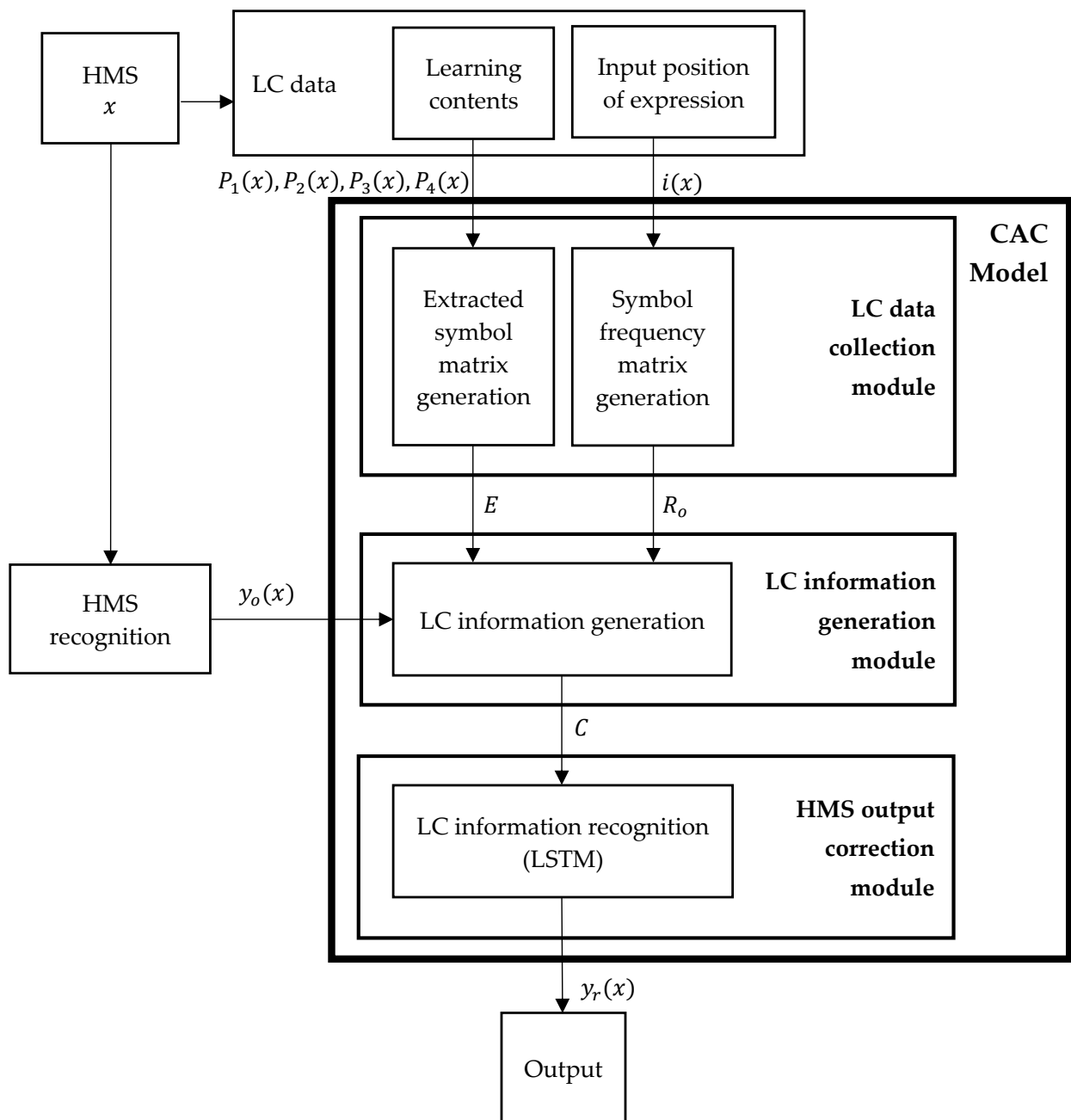


Figure 5. CAC model using LC information to recognize HMS.

First, the LC data collection module collects LC data related to the HMS, such as symbols included in the learning contents and the input position of the expression, from the e-learning system. Next, the LC information generation module converts the collected LC data into LC information so that it can be used in the artificial neural network. Finally, the HMS output correction module recognizes the LC information through an artificial neural network based on the LSTM and corrects the incorrect HMS output to improve the recognition accuracy.

4.2. LC Data Collection Module

The LC data collection module collects four learning parts, which are $P_1(x)$, $P_2(x)$, $P_3(x)$, and $P_4(x)$, and input position, which is $i(x)$, for the HMS x from the e-learning system.

4.2.1. Extracted Symbol Matrix Generation

$E_n(x)$ ($1 \leq n \leq 4$) is a set of extracted symbols for each learning part $P_n(x)$.

$$E_n(x) = \bigcup_{\varphi_k \in F(P_n(x))} S(\varphi_k) \tag{14}$$

Therefore, $E_1(x)$, $E_2(x)$, $E_3(x)$, and $E_4(x)$ are sets of extracted symbols within the learning topic, the question, the solving processes, and the correct answers related to the question that the learner is solving when symbol x is input, respectively.

The extracted symbol matrix E is a matrix containing information about the symbols included in the expressions of each learning part. A $4 \times n_D$ matrix E is obtained as follows:

$$E = \begin{bmatrix} e_{11} & e_{12} & \cdots & e_{1n_D} \\ e_{21} & e_{22} & \cdots & e_{2n_D} \\ e_{31} & e_{32} & \cdots & e_{3n_D} \\ e_{41} & e_{42} & \cdots & e_{4n_D} \end{bmatrix} \tag{15}$$

where $e_{ni} = \begin{cases} 0, & d_i \notin E_n \\ 1, & d_i \in E_n \end{cases}$, and symbol d_i means the i th symbol of the symbol list D .

4.2.2. Symbol Frequency Matrix Generation

Assuming that learners input expressions related to the learning contents during mathematic learning, symbols of learning contents tend to be frequently used in expressions input by learners. However, not all symbols have the same frequency. It is therefore necessary to obtain symbol frequency rates, which indicate how often symbols in one learning part are used in another learning part, and to reflect these in adjusting the HMS output. Symbol frequency rates can be obtained from learning contents stored in an e-learning system using the statistical probability of how much the symbols of each learning part match those of the other learning parts.

For a symbol s and a learning part set X , $\bar{L}_{Prs}(s, X)$ and $\bar{L}_{Ans}(s, X)$ are defined as sets of solving processes and answers, respectively, related to all learning parts containing s within X .

$$\bar{L}_{Prs}(s, X) = \bigcup_{P_k \in \bar{P}(s, X)} L_3(P_k) \tag{16}$$

$$\bar{L}_{Ans}(s, X) = \bigcup_{P_k \in \bar{P}(s, X)} L_4(P_k) \tag{17}$$

For example, $\bar{L}_{Prs}(s, U_2)$ is the set of solving processes related to all questions containing symbol s , and $\bar{L}_{Ans}(s, U_3)$ is the set of answers related to all solving processes containing symbol s .

For a symbol s and a learning part set X , the symbol frequency rate $f_{Prs}(s, X)$ is defined as the frequency at which expressions containing s are used in the solving processes related to learning parts including s in X and calculated as follows.

$$f_{Prs}(s, X) = \frac{\sum_{P_l \in \bar{P}(s, X)} n(F(P_l))}{\sum_{P_k \in \bar{L}_{Prs}(s, X)} n(F(P_k))} \tag{18}$$

where $n(A)$ means the number of all elements in set A . For example, $f_{Prs}(s, U_2)$ is, in all solving processes related to questions containing s , the number of expressions containing s , divided by the number of all expressions.

Similar to $f_{PrC}(s, X)$, for symbol s and learning part set X , the symbol frequency rate $f_{Ans}(s, X)$ is defined as the frequency at which expressions containing s are used in the correct answers related to learning parts including s in X and is calculated as follows.

$$f_{Ans}(s, X) = \frac{\sum_{P_l \in \bar{P}(s, \bar{L}_{Ans}(s, X))} n(F(P_l))}{\sum_{P_k \in \bar{L}_{Ans}(s, X)} n(F(P_k))} \tag{19}$$

Using the symbol frequency rates of symbols in each learning part, the symbol frequency matrices R_{PrC} and R_{Ans} with a size of $4 \times n_D$ can be obtained. R_{PrC} represents the symbol frequency rates of symbols in the solving process when they are used in each learning part, and R_{Ans} represents the symbol frequency rates of symbols in the correct answer when they are used in each learning part, as shown in Equations (20) and (21):

$$R_{PrC} = \begin{bmatrix} f_{PrC}(d_1, U_1) & f_{PrC}(d_2, U_1) & \cdots & f_{PrC}(d_{n_D}, U_1) \\ f_{PrC}(d_1, U_2) & f_{PrC}(d_2, U_2) & \cdots & f_{PrC}(d_{n_D}, U_2) \\ f_{PrC}(d_1, U_3) & f_{PrC}(d_2, U_3) & \cdots & f_{PrC}(d_{n_D}, U_3) \\ f_{PrC}(d_1, U_4) & f_{PrC}(d_2, U_4) & \cdots & f_{PrC}(d_{n_D}, U_4) \end{bmatrix} \tag{20}$$

$$R_{Ans} = \begin{bmatrix} f_{Ans}(d_1, U_1) & f_{Ans}(d_2, U_1) & \cdots & f_{Ans}(d_{n_D}, U_1) \\ f_{Ans}(d_1, U_2) & f_{Ans}(d_2, U_2) & \cdots & f_{Ans}(d_{n_D}, U_2) \\ f_{Ans}(d_1, U_3) & f_{Ans}(d_2, U_3) & \cdots & f_{Ans}(d_{n_D}, U_3) \\ f_{Ans}(d_1, U_4) & f_{Ans}(d_2, U_4) & \cdots & f_{Ans}(d_{n_D}, U_4) \end{bmatrix} \tag{21}$$

In the LC data collection module, the symbol frequency matrix R_o is selected as follows by reflecting the input position $i(x)$ of the expression to adjust the recognition output of the input HMS x efficiently.

$$R_o = \begin{cases} R_{PrC}, & i(x) = 0 \text{ (when } x \text{ is in an expression of a solving process)} \\ R_{Ans}, & i(x) = 1 \text{ (when } x \text{ is in an expression of an answer)} \end{cases} \tag{22}$$

4.3. LC Information Generation Module

The LC information generation module receives the extracted symbol matrix E and symbol frequency matrix R_o from the LC data collection module and generates the expected symbol matrix R , which is the LC information.

The LC information used in the CAC model is the expected symbol list for the input HMS and the expected value of each expected symbol. For an HMS, the expected symbol is defined as a symbol with the probability to be the correct one, and the expected value means the probability.

The learner tends to use symbols related to the learning contents of each learning part when inputting the expression. Therefore, in the CAC model, the extracted symbols of each learning part are considered the expected symbols, and the expected value of each symbol is set to the symbol frequency rate of this. Therefore, from the extracted symbol matrix E (Equation (15)) generated through the extracted symbols of each learning part and the symbol frequency rate matrix R_o (Equation (22)) generated through the symbol frequency rate and the input position of the equation, the expected symbol matrix R with a size of $4 \times n_D$, which is the LC information, is calculated as follows:

$$R = R_o \otimes E = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n_D} \\ r_{21} & r_{22} & \cdots & r_{2n_D} \\ r_{31} & r_{32} & \cdots & r_{3n_D} \\ r_{41} & r_{42} & \cdots & r_{4n_D} \end{bmatrix} \tag{23}$$

where \otimes stands for element-wise multiplication of the matrices. In addition, each element r_{hi} of the expected symbol matrix R is the expected value of the symbol $d_i \in D$ obtained from each learning part (t).

4.4. HMS Output Correction Module and Output

The HMS output correction module receives the HMS output $y_o(x)$ and the expected symbol matrix R obtained from the LC information generation module, which are merged as follows and transformed into LC information matrix C with a size of $5 \times n_D$.

$$C = \begin{bmatrix} y_o(x) \\ R \end{bmatrix} = \begin{bmatrix} p_1^o & p_2^o & \cdots & p_{n_D}^o \\ r_{11} & r_{12} & \cdots & r_{1n_D} \\ r_{21} & r_{22} & \cdots & r_{2n_D} \\ r_{31} & r_{32} & \cdots & r_{3n_D} \\ r_{41} & r_{42} & \cdots & r_{4n_D} \end{bmatrix} \quad (24)$$

The 2nd, 3rd, 4th, and 5th row of matrix C , which are the rows of the expected symbol matrix R , are referred to as sub-contextual information 1, 2, 3, and 4 respectively. To apply these to the HMS output adjustment, one aspect must first be solved, i.e., the problem regarding how much weight each sub-contextual piece of information must have in the coordination of the LC information to achieve the best results. It is difficult to obtain an optimal weight, and even if it is obtained, the list of symbols used for each learning part and the symbol frequency rate are different depending on the learning range and learning contents; therefore, the values also change when the learning conditions change. In the CAC model, a complex algorithm for obtaining these variable weights is implemented using an artificial neural network.

Therefore, the role of the artificial neural network used to recognize the LC information is to improve the accuracy of the HMS output by assigning optimal weights to each sub-contextual information. To this end, in the artificial neural network, the HMS output should be related to all sub-contextual information, and each weight should be applied appropriately. However, in matrix C , sub-contextual information is sequentially listed following the HMS output; therefore, an appropriate method for linking the HMS output with all sub-contextual information is required. To efficiently solve this problem, in this paper, LSTM was applied as shown in Figure 6. The parameters of LSTM play the role of weight to be applied to each element of sub-contextual information (x_2, x_3, x_4 , and x_5) to be calculated with HMS output (x_1). In detail, appropriate weights between HMS output and all sub-contextual information are calculated through the cell state (c_i) responsible for long-term memory in LSTM. In addition, the relationship between sub-contextual information through the hidden state (h_i) responsible for short-term memory along with cell state is also reflected in the weight. Matrix C is transformed into the context-applied output $y_r(x) = [p_1^r \ p_2^r \ \cdots \ p_{n_D}^r]$, which is a row vector with a size of $1 \times n_D$, through the artificial neural network constructed using LSTM.

Finally, considering the context-applied output $y_r(x)$, θ^r , which is the index of the element with the maximum value, is obtained. That is, $\theta^r = \underset{1 \leq i \leq n_D}{\operatorname{argmax}}(p_i^r)$. As a result, the symbol $d_{\theta^r} \in D$ with index θ^r becomes the final output of the CAC model.

The definitions of all symbols and functions in Section 4 are summarized in Appendix B.

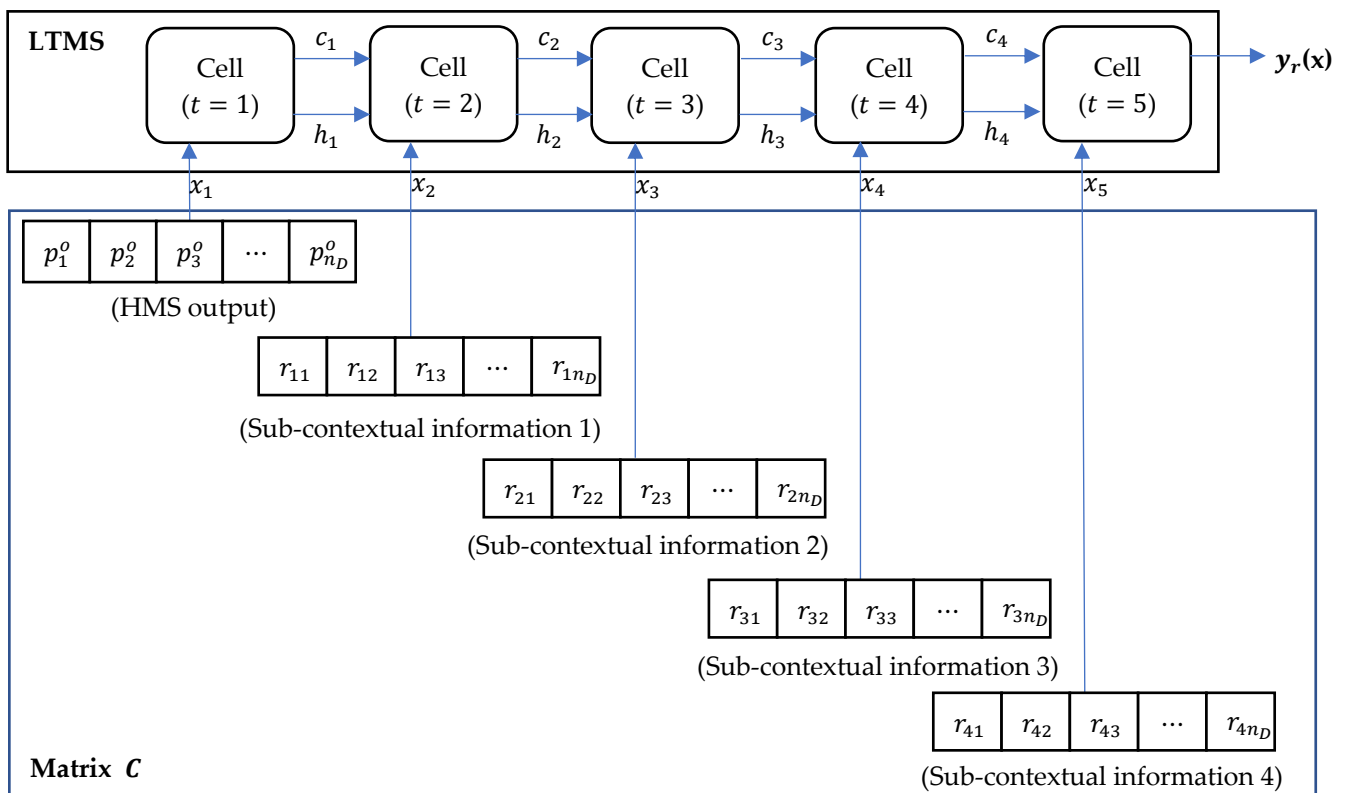


Figure 6. Application of LSTM to LC information recognition. t is the timestep; x_t, c_t, h_t are the input vector, cell state, and hidden state, respectively, at timestep t .

5. Experiment

5.1. Experiment Environment

In this paper, the results of HMS recognition were compared according to whether the LC information was applied using a dataset configured similarly to the actual learning conditions. To this end, units of rational numbers, the calculation of the monomials, and the calculation of the polynomials in a mathematics workbook [23] for middle school students were set up as experimental targets. Like the composition of learning contents in this paper, each question in the workbook is related to a topic, a solving process, and a correct answer. The learning contents within the units consisted of 11 topics and 557 questions related to those topics, and a total of 50 symbols were used. The list of all symbols used in these units is provided in Table 3.

As discussed above, in the analyzed data, it can be confirmed that there is a difference in the symbol frequency rate between the solving processes and the correct answers. Assuming that the learners studying these units write expressions that are similar to the model-solving processes, as shown in Appendices C–F, the extracted symbols from the learning topics, questions, and solving processes are used more repeatedly in the expressions of the solving processes than in the expressions of the answer.

Accordingly, as shown in Table 4, 89,477 data points for 50 symbols among the datasets of the 2019 CROHME online symbol recognition task (Task 1a) were used for the experiment. Among them, 81,265 were training data, and 8212 were test data.

Table 3. Symbols used in the experiment.

Index	Symbol	Latex	Index	Symbol	Latex	Index	Symbol	Latex
1	7	7	18	b	b	35	c	c
2	1	1	19	a	a	36	A	A
3	\times	\times	20	F	F	37	B	B
4	t	t	21	C	C	38	[[
5	–	-	22	5	5	39]]
6	2	2	23	9	9	40	<	\lt
7	x	x	24	8	8	41	L	L
8	=	=	25	π	\pi	42	h	h
9	n	n	26	d	d	43	E	E
10	y	y	27	\div	\div	44	V	V
11	z	z	28	0	0	45	s	s
12))	29	g	g	46	q	q
13	((30	p	p	47	l	l
14	+	+	31	r	r	48	v	v
15	6	6	32	m	m	49	M	M
16	3	3	33	\leq	\leq	50	I	I
17	4	4	34	.	.			

Table 4. Composition of dataset used in the experiment.

Purpose	Training Set (HMS Recognition and LC Information Recognition)	Testing Set	Total
Number of data points	81,265	8212	89,477

However, the CROHME dataset did not contain the LC data required for this experiment. Two methods can be considered to arbitrarily match the learning contents of the workbook and the CROHME dataset: (1) a method of allocating the symbols of the CROHME dataset to the LC data constructed from the learning contents, and (2) a method of allocating LC data to the symbols of the CROHME dataset similarly to the learning contents. In the method of (1), the same LC data as the actual learning contents are composed, but many of the symbols of the CROHME dataset are omitted or duplicated. On the other hand, the method of (2) uses all symbols of the CROHME dataset without omission or duplication, but the LC data do not completely match the learning content. In this paper, method (2) was used as follows.

- **Input position:** 16,821 data points of the training set and 1714 data points of the test set, randomly selected according to the ratio of pre-investigated statistics, were set to the symbols of the expression in the answer parts; that is, their input positions were set to answer parts. The others' input positions were set to solving processes.
- **Extracted symbols:** As shown in Table 5, for a given symbol, there are 16 cases (00 to 15) of a method of designating extracted symbols of the four learning parts, depending on which learning part contains the symbol for data where the symbol is the correct one. Similarly, there are 16 cases for data where the symbol is not the correct one as well. Therefore, all data can be divided into 32 cases for each symbol. For each symbol, we randomly portioned the entire CROHME dataset according to the 32 ratios calculated from the number of symbols in the learning contents to make the setting similar to the actual learning environment. As can be seen in Table 6,

which compares the ratio of extracted symbols for symbol ‘2’, in all cases in Table 5, we matched the ratios of extracted symbols assigned to the CROHME dataset to the ratios of the symbols in the learning contents. As a result, the symbol frequency rates of the CROHME dataset became the same as the symbol frequency rates of the learning contents.

Table 5. Classification of LC data according to whether the symbol is included in each learning part.

Learning Part	Whether to Include the Symbol (Case 00 to 15) ¹															
	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
Learning topic	×	×	×	×	×	×	×	×	○	○	○	○	○	○	○	○
Question	×	×	×	×	○	○	○	○	×	×	×	×	○	○	○	○
Solving process	×	×	○	○	×	×	○	○	×	×	○	○	×	×	○	○
Answer	×	○	×	○	×	○	×	○	×	○	×	○	×	○	×	○

¹ ○: the symbol is included in the learning part, ×: the symbol is not included in the learning part.

Table 6. Classification of LC data for symbol ‘2’ and the number of data.

Case	Number of Data (Correct Symbol Is ‘2’)			Number of Data (Correct Symbol Is Not ‘2’)		
	Workbook	Train Dataset	Test Dataset	Workbook	Train Dataset	Test Dataset
00	58 (1.8%)	95 (1.8%)	10 (1.9%)	183 (10.4%)	6150 (10.4%)	620 (10.4%)
01	41 (1.2%)	66 (1.2%)	7 (1.3%)	62 (3.5%)	2084 (3.5%)	210 (3.5%)
02	136 (4.1%)	219 (4.1%)	22 (4.1%)	58 (3.3%)	1949 (3.3%)	196 (3.3%)
03	111 (3.4%)	179 (3.4%)	18 (3.4%)	41 (2.3%)	1378 (2.3%)	139 (2.3%)
04	244 (7.4%)	394 (7.4%)	40 (7.4%)	324 (18.4%)	10,888 (18.4%)	1097 (18.4%)
05	157 (4.8%)	253 (4.8%)	26 (4.8%)	179 (10.2%)	6015 (10.2%)	606 (10.2%)
06	980 (29.8%)	1581 (29.8%)	160 (29.8%)	244 (13.9%)	8200 (13.9%)	826 (13.9%)
07	1042 (31.7%)	1681 (31.7%)	170 (31.7%)	157 (8.9%)	5276 (8.9%)	532 (8.9%)
08	13 (0.4%)	21 (0.4%)	2 (0.4%)	227 (12.9%)	7628 (12.9%)	769 (12.9%)
09	4 (0.1%)	6 (0.1%)	1 (0.2%)	47 (2.7%)	1579 (2.7%)	159 (2.7%)
10	11 (0.3%)	18 (0.3%)	2 (0.4%)	13 (0.7%)	437 (0.7%)	44 (0.7%)
11	9 (0.3%)	15 (0.3%)	1 (0.2%)	4 (0.2%)	134 (0.2%)	14 (0.2%)
12	58 (1.8%)	94 (1.8%)	9 (1.7%)	56 (3.2%)	1882 (3.2%)	190 (3.2%)
13	48 (1.5%)	77 (1.5%)	8 (1.5%)	59 (3.4%)	1983 (3.4%)	200 (3.4%)
14	194 (5.9%)	313 (5.9%)	32 (6.0%)	58 (3.3%)	1949 (3.3%)	196 (3.3%)
15	178 (5.4%)	287 (5.4%)	29 (5.4%)	48 (2.7%)	1613 (2.7%)	163 (2.7%)
Total	3284	5299	537	1760	59,145	5961

Table 7 shows samples in which input position and extracted symbols are arbitrarily assigned to data points.

In addition, the TAP model was used to recognize the HMS of all datasets. TAP is the model used by the USTC-iFLYTEK team and achieved the best results for the online handwritten mathematical expression recognition task (Task 1) at 2019 CROHME, and its source code is open for use in other studies [15].

As discussed in Section 4.4, the artificial neural network used in the HMS output correction module of the CAC model should be able to grasp the relationship between the HMS output and all sub-contextual information sequentially arranged in the LC information. Therefore, the LSTM was used for the artificial neural network. For efficient training and an adjustment of the outputs, dropout [24], fully connected [24], and softmax [25] layers were added to the artificial neural network, as shown in Table 8. The output dimension of each layer was set to 50, which is the total number of symbols used in this paper. To prevent an overfitting, the dropout ratio was set to 0.5.

Table 7. Sample data points assigned LC data.



CROME Dataset		LC Data				
HMS	Correct Symbol	Extracted Symbols			Correct Answer	Input Position
		Learning Topic	Question	Solving Process		
	<i>m</i>	.	2	2 = 3	1 3	Solving process
		+	7			
		−	9			
		[=			
]	<i>m</i>			
	7	−	2	2	0 3 <i>b</i>	Solving process
		(3	3		
		[8	8		
]	<i>b</i>	=		
			<i>x</i>	<i>b</i>		

Table 8. Configuration of LC information recognition based artificial neural network.

No.	Layer	Setting
1	LSTM	Output dimension = 50
2	Dropout	Rate = 0.5
3	Fully connected (dense)	Output dimension = 50
4	Softmax	Output dimension = 50

5.2. Training and Testing

Two groups were used in the experiment. As shown in Table 9, in experiment group I, the TAP model was trained using 81,265 HMS data points. The TAP was tested at every epoch on the testing set. Subsequently, the entire training set was recognized again through the trained TAP model to obtain the HMS output dataset for experiment group II. In experiment group II, the CAC model was trained using LC data constructed by the method discussed in Section 5.1, along with the obtained HMS output dataset. At this time, 24,379 data points, which is 30% of the total training set, were used as the validation set. The CAC model of experiment group II was tested at every epoch on the validation set. The training of each experiment group ended before the decrease in recognition accuracy.

Table 9. Training set used for each experimental group.

Experimental Group	Dataset (81,265 Data Points)		Artificial Neural Network to Train
	Training Set	Validation Set	
I	HMS (81,265 data points)	-	TAP
II	HMS output data obtained using the model of experimental group I after training (56,886 data points)	24,379 data points (30%)	CAC
	LC data (56,886 data points)		

The model evaluation test measured the accuracy of the same testing set in both experiment groups I and II. The HMS recognition results of the testing set obtained using the trained TAP model of experiment group I were used as the HMS output data for experiment group II.

5.3. Results and Discussion

As shown in Table 10, the results for experiment group I showed that the accuracy of the TAP model, which recognized only the shape of the HMS, was 93.22%. On the other hand, the results of experiment group II showed that the recognition accuracy of the CAC model, which adjusts the HMS outputs of the TAP model by applying the LC information, was 97.15%, which was 3.93% higher than that of experiment group I. These results indicate that the LC information recognition improves the accuracy of the HMS recognition results.

Table 10. Experiment results.

Experimental Group	Model	Test Subject	Accuracy (Number of Symbols)		
			Solving Processes (6498)	Answers (1714)	Solving Process + Answers (8212)
I	TAP	Recognition of HMS	93.20% (6056)	93.29% (1599)	93.22% (7655)
II	TAP + CAC	Recognition of HMS outputs and LC data	96.48% (6269)	99.71% (1709)	97.15% (7978)

The recognition accuracies of the TAP model for HMS in solution processes and answers are similar, at 93.20% and 93.29%, respectively, while the recognition accuracies of the CAC model differ by 96.48% and 99.71%, respectively. This means that the effect of using LC information in the solution processes is different from that in the correct answers.

More specifically, the recognition results of experiment groups I and II were compared, as shown in Table 11. In total, 404 data points, which were the symbols with an ambiguous representation misrecognized by the TAP model, were accurately adjusted through the CAC model. Conversely, 81 data points that were properly recognized in the TAP model were incorrectly recognized as they went through the CAC model; however, they accounted for 0.99% of the total data, which is a relatively small number.

Since HMS recognition and LC information recognition processes are independent of each other, not only the TAP model used in the experiment but also any recognition model that outputs the probability of each symbol as a result of HMS recognition can be linked with the CAC model. In addition, no matter which model it is interlocked with, the CAC model will be able to perform.

In the experiment, the actual LC data of e-learning systems could not be tested. In addition, there is a limitation in that data of various learning content ranges could not be tested. These are because sufficient LC data paired with HMS could not be obtained. Future work will further refine LC data and experiment with a wide range of data. In addition, some expressions entered by learners in solving processes and answers might not match model-solving processes and correct answers, respectively. If learners use symbols different from the ones proposed in the learning contents, LC data could worsen the recognition performance of the CAC model. The recognition method of HMS entered by learners inconsistent with LC data is a task to be studied in the future.

In this paper, a simple LSTM model is used as a method for recognizing learning context information in the CAC model. However, methods using more elaborately set LSTM or other artificial intelligence models (such as BLSTM) need to be studied in the future.

Table 11. Corrected and missed symbols using the CAC model.

Recognition Result		Number of Data	Symbols with Recognition Results	
TAP	TAP + CAC		Output of TAP → Output of CAC (Number of Data)	HMS
Error	Correct	404 (4.92%)	$\times \rightarrow x$ (47)	
			$C \rightarrow c$ (20)	
			$x \rightarrow \times$ (17)	
			$t \rightarrow +$ (15)	
			⋮	⋮
Correct	Error	81 (0.99%)	$x \rightarrow \times$ (22)	
			$1 \rightarrow)$ (5)	
			$a \rightarrow 9$ (4)	
			$2 \rightarrow =$ (4)	
			⋮	⋮

6. Conclusions

An e-learning system should support learners who learn mathematics to write mathematical expressions freely. However, handwritten mathematical expressions contain many ambiguous symbols. Most existing studies have mainly used the shape of the symbol to recognize the HMS. This method has limitations in terms of accurately predicting ambiguous symbols, even for humans.

In this paper, the CAC model was designed to use LC data and improve the results of existing studies on e-learning environments. In the CAC model, sufficient LC information was generated using data outside the expressions, i.e., LC data that are relatively indirectly related to the HMS. In the process of using LC information to adjust the output of the HMS recognition, the optimal weight is applied to each sub-contextual piece of LC information through an artificial neural network.

In the experiment, the existing and CAC models were trained and tested on a dataset similar to the actual learning environment. The results showed that the CAC model

corrected the misrecognized results of the existing model, and the recognition accuracy improved. Therefore, it was found that the use of LC information proposed in this paper has a positive effect on improving the accuracy of HMS recognition.

Author Contributions: Conceptualization, S.-B.B. and J.-G.S.; methodology, S.-B.B., J.-G.S. and J.-S.P.; software, S.-B.B.; validation, S.-B.B., J.-G.S. and J.-S.P.; formal analysis, S.-B.B.; investigation, S.-B.B.; resources, S.-B.B.; data curation, S.-B.B.; writing—original draft preparation, S.-B.B.; writing—review and editing, S.-B.B., J.-G.S. and J.-S.P.; visualization, S.-B.B. and J.-G.S.; supervision, J.-G.S. and J.-S.P.; project administration, J.-G.S. and J.-S.P.; funding acquisition, J.-S.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Definitions of Symbols and Functions in Section 3.

Section	Symbol/ Function	Definition	Equation
Section 3.1	U	the universal set of learning parts in the e-learning system	(1)
	U_1	the set of all learning topics	(2)
	U_2	the set of all questions	(3)
	U_3	the set of all solving process	(4)
	U_4	the set of all correct answers	(5)
	$L_1(P)$	the learning part set of learning topics related to a learning part P	(6)
	$L_2(P)$	the learning part set of questions related to a learning part P	(7)
	$L_3(P)$	the learning part set of solving processes related to a learning part P	(8)
	$L_4(P)$	the learning part set of correct answers related to a learning part P	(9)
	$F(P)$	the set of all expressions included in a learning part P	(10)
Section 3.2	$S(\varphi)$	the set of all extracted symbols in an expression φ	(11)
	$\bar{P}(s, X)$	the set of all learning parts containing a symbol s within a learning part set X	(12)
Section 3.3	$P_1(x)$	the learning topic that the learner is studying when HMS x is input	(13)
	$P_2(x)$	the question that the learner is solving when HMS x is input	
	$P_3(x)$	the solving process of the question that the learner is solving when HMS x is input	
	$P_4(x)$	the correct answer of the question that the learner is solving when HMS x is input	
	$i(x)$	the value indicating which type of learning part when HMS x is input	
	D	the ordered list of all symbols used as an output of the HMS recognition	
	n_D	the total number of symbols in symbol list D	
	$d_i \in D$	the i th symbol of symbol list D	
	$y_o(x)$	the HMS output, which is the recognition output of HMS x	
$y_r(x)$	the context – applied output, which is the adjusted output of the HMS output $y_o(x)$		

Appendix B

Table A2. Definitions of Symbols and Functions in Section 4.

Section	Symbol/ Function	Definition	Equation
Section 4.2	$E_n(x)$	the set of extracted symbols for each learning part $P_n(x)$	(14)
	E	the matrix containing information about the symbols included in the expressions of each learning part	(15)
		the set of solving processes related to all learning parts containing a symbol s within a learning part set X	(16)
	$\bar{L}_{Ans}(s, X)$	the set of answers related to all learning parts containing a symbol s within a learning part set X	(17)
	$f_{Prc}(s, X)$	the frequency at which expressions containing a symbol s are used in the solving processes related to learning parts including a symbol s in a learning part set X	(18)
	$f_{Ans}(s, X)$	the frequency at which expressions containing a symbol s are used in the correct answers related to learning parts including a symbol s in a learning part set X	(19)
	R_{Prc}	the matrix that represents symbol frequency rates of symbols in the solving process when they are used in the learning topics, the questions, the solving processes, and the correct answers	(20)
	R_{Ans}	the matrix that represents symbol frequency rates of symbols in the correct answer when they are used in the learning topics, the questions, the solving processes, and the correct answers	(21)
	R_o	the matrix selected from R_{Prc} and R_{Ans} by reflecting the input position $i(x)$ of the expression	(22)
	Section 4.3	R	the expected symbol matrix, which is calculated as $R_o \otimes E$
Section 4.4	C	the LC information matrix, which is a merge of the HMS output y_o and the expected symbol matrix R	(24)
	θ^r	the index of the element with the maximum value in the context-applied output $y_r(x)$	

Appendix C

Table A3. Sample Symbol Frequency Rates in Solving Processes and Correct Answers for Symbols Extracted from Learning Topics.

Extracted Symbols of Learning Topics	Symbol Frequency Rate		Extracted Symbols of Learning Topics	Symbol Frequency Rate			
	Solving Process	Correct Answer		Solving Process	Correct Answer		
Numbers	0	133/243 (55%)	51/100 (51%)	—	682/1326 (51%)	223/554 (40%)	
	1	145/243 (60%)	41/100 (41%)	—(fraction)	423/1326 (32%)	123/554 (22%)	
	2	126/243 (52%)	39/100 (39%)	Signs	(388/1083 (36%)	10/454 (2%)
	5	84/243 (35%)	35/100 (35%))	388/1083 (36%)	10/454 (2%)	
				⋮	⋮	⋮	

Appendix D

Table A4. Sample Symbol Frequency Rates in Solving Processes and Correct Answers for Symbols Extracted from Questions.

Extracted Symbols of Questions	Symbol Frequency Rate		Extracted Symbols of Questions	Symbol Frequency Rate	
	Solving Process	Correct Answer		Solving Process	Correct Answer
Numbers	2	816/1048 (78%)	Uppercases	A	41/115 (36%)
	3	535/821 (65%)		B	28/92 (30%)
	1	406/606 (67%)		C	21/48 (44%)
	4	279/517 (54%)		S	29/40 (73%)
	∴	∴		∴	∴
Lowercases	x	474/667 (71%)	Signs	–	553/718 (77%)
	a	331/436 (76%)		(265/664 (40%)
	y	257/394 (65%))	265/661 (40%)
	b	214/314 (68%)		+	424/644 (66%)
	∴	∴		∴	∴

Appendix E

Table A5. Sample Frequency Rates in Different Solving Processes and Correct Answers for Symbols Extracted from Such Processes.

Extracted Symbols of Solving Processes	Symbol Frequency Rate		Extracted Symbols of Solving Processes	Symbol Frequency Rate	
	Solving Process ¹	Correct Answer		Solving Process ¹	Correct Answer
Numbers	2	1744/2367 (74%)	Uppercases	A	270/488 (55%)
	1	980/1638 (60%)		B	96/259 (37%)
	3	928/1542 (60%)		C	46/137 (34%)
	4	492/1011 (49%)		S	68/82 (83%)
	∴	∴		∴	∴

Table A5. Cont.

Extracted Symbols of Solving Processes	Symbol Frequency Rate		Extracted Symbols of Solving Processes	Symbol Frequency Rate		
	Solving Process ¹	Correct Answer		Solving Process ¹	Correct Answer	
Lowercases	x	886/1297 (68%)	313/524 (60%)	=	2972/3269 (91%)	73/1189 (6%)
	a	560/786 (71%)	232/337 (69%)	−	1350/1750 (77%)	413/679 (61%)
	b	390/620 (63%)	146/224 (65%)	+	878/1391 (63%)	223/529 (42%)
	y	318/558 (57%)	155/260 (60%)	−(fraction)	710/1097 (65%)	175/422 (41%)
	⋮	⋮	⋮	⋮	⋮	⋮

¹ Only cases with two or more expressions in the solving process of one question were counted.

Appendix F

Table A6. Sample Symbol Frequency Rates in Solving Processes and Correct Answers for Symbols Extracted from Correct Answers.

Extracted Symbols of Correct Answers	Symbol Frequency Rate		Extracted Symbols of Correct Answers	Symbol Frequency Rate		
	Solving Process	Correct Answer		Solving Process	Correct Answer	
Numbers	2	514/659 (78%)	303/303 (100%)	S	26/33 (79%)	13/13 (100%)
	1	267/494 (54%)	196/196 (100%)	A	2/6 (33%)	2/2 (100%)
	3	224/391 (57%)	160/160 (100%)	V	4/6 (67%)	3/3 (100%)
	4	120/287 (42%)	128/128 (100%)	C	4/4 (100%)	2/2 (100%)
	⋮	⋮	⋮	⋮	⋮	⋮
Lowercases	x	313/350 (89%)	173/173 (100%)	−	413/498 (83%)	223/223 (100%)
	a	232/257 (90%)	134/134 (100%)	−(fraction)	175/279 (63%)	123/123 (100%)
	y	155/178 (87%)	96/96 (100%)	+	223/261 (85%)	133/133 (100%)
	b	146/172 (85%)	91/91 (100%)	=	73/79 (92%)	33/33 (100%)
	⋮	⋮	⋮	⋮	⋮	⋮

References

1. Babli, M.; Rincon, J.A.; Onaindia, E.; Carrascosa, C.; Julian, V. Deliberative context-aware ambient intelligence system for assisted living homes. *Hum.-Cent. Comput. Inf. Sci.* **2021**, *11*, 19. [CrossRef]
2. Khowaja, S.A.; Yahya, B.N.; Lee, S.L. CAPHAR: Context-aware personalized human activity recognition using associative learning in smart environments. *Hum.-Cent. Comput. Inf. Sci.* **2020**, *10*, 35. [CrossRef]
3. Chan, K.; Yeung, D. Mathematical expression recognition: A survey. *Int. J. Doc. Anal. Recogn.* **2000**, *3*, 3–15. [CrossRef]

4. Chan, C.K. Stroke extraction for offline handwritten mathematical expression recognition. *IEEE Access* **2020**, *8*, 61565–61575. [[CrossRef](#)]
5. Zhang, T. New architectures for handwritten mathematical expressions recognition. In *Image Processing*; Université de Nantes: Nantes, France, 2017.
6. Zhang, J.; Du, J.; Zhang, S.; Liu, D.; Hu, Y.; Hu, J.; Wei, S.; Dai, L. Watch, attend and parse: An end-to-end neural network based approach to handwritten mathematical expression recognition. *Pattern Recognit.* **2017**, *71*, 196–206. [[CrossRef](#)]
7. Miller, E.G.; Viola, P.A. Ambiguity and constraint in mathematical expression recognition. *Am. Assoc. Artif. Intell.* **1998**, 784–791. [[CrossRef](#)]
8. Kosmala, A.; Rigoll, G. On-line handwritten formula recognition using statistical methods. *Fourteenth Int. Conf. Pattern Recognit.* **1998**, *2*, 1306–1308. [[CrossRef](#)]
9. Chou, P.A. Recognition of Equations Using a Two-Dimensional Stochastic Context-Free Grammar. *Vis. Commun. Image Process. IV* **1989**, *1199*, 852–863. [[CrossRef](#)]
10. Álvaro, F.; Sánchez, J.A.; Benedí, J.M. An integrated grammar-based approach for mathematical expression recognition. *Pattern Recognit.* **2016**, *51*, 135–147. [[CrossRef](#)]
11. Zhelezniakov, D.; Zaytsev, V.; Radyvonenko, O. Acceleration of Online Recognition of 2D Sequences using Deep Bidirectional LSTM and Dynamic Programming. *Adv. Comput. Intell.* **2019**, *11507*, 438–449. [[CrossRef](#)]
12. Naik, S.A.; Metkewar, P.S.; Mapari, S.A. Recognition of ambiguous mathematical characters within mathematical expressions. Symbiosis Institute of Computer Studies and Research. In Proceedings of the 2017 International Conference on Electrical Computer and Communication Technologies, Coimbatore, India, 22–24 February 2017; pp. 1–4. [[CrossRef](#)]
13. Álvaro, F.; Sánchez, J.A.; Benedí, J.M. Offline Features for Classifying Handwritten Math Symbols with Recurrent Neural Networks. In Proceedings of the 2014 22nd International Conference on Pattern Recognition, Stockholm, Sweden, 24–28 August 2014; pp. 2944–2949. [[CrossRef](#)]
14. Mahdavi, M.; Zanibbi, R.; Mouch'ere, H.; Viard-Gaudin, C.; Garain, U. ICDAR 2019 CROHME + TFD: Competition on recognition of handwritten mathematical expressions and typeset formula detection. In Proceedings of the 2019 International Conference on Document Analysis and Recognition, Sydney, NSW, Australia, 20–25 September 2019. [[CrossRef](#)]
15. Zhang, J.; Du, J.; Dai, L. Track, attend and parse (TAP): An end-to-end framework for online handwritten mathematical expression recognition. *IEEE Trans. Multimed.* **2019**, *21*, 221–233. [[CrossRef](#)]
16. Degtyarenko, I.; Radyvonenko, O.; Bokhan, K.; Khomenko, V. Text/shape classifier for mobile applications with handwriting input. *Int. J. Doc. Anal. Recogn.* **2016**, *19*, 369–379. [[CrossRef](#)]
17. Wu, J.; Yin, F.; Zhang, Y.; Zhang, X.; Liu, C. Image-to-markup generation via paired adversarial learning. In *Machine Learning and Knowledge Discovery in Databases*; Springer International Publishing: Cham, Switzerland, 2019; pp. 18–34.
18. Le, A.; Nakagawa, M. A system for recognizing online handwritten mathematical expressions by using improved structural analysis. *Int. J. Doc. Anal. Recogn.* **2016**, *19*, 305–319. [[CrossRef](#)]
19. Kim, H.C.; Lee, S.W. Document summarization model based on general context in RNN. *J. Inf. Process. Syst.* **2019**, *15*, 1378–1391. [[CrossRef](#)]
20. Om, K.; Boukoros, S.; Nugaliyadde, A.; McGill, T.; Dixon, M.; Koutsakis, P.; Wong, K. Modelling email traffic workloads with RNN and LSTM models. *Hum.-Cent. Comput. Inf. Sci.* **2020**, *10*, 1–16. [[CrossRef](#)]
21. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
22. Olah, C. Understanding LSTM Networks. 2015. Available online: <https://colah.github.io/posts/2015-08-Understanding-LSTMs> (accessed on 17 January 2022).
23. Yang, T. *Concept Plus Type Middle School Mathematics 2-1*; Concept Volume; Visang Education: Seoul, Korea, 2011.
24. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
25. Wood, T. Softmax Function. 2019. Available online: <https://deepai.org/machine-learning-glossary-and-terms/softmax-layer> (accessed on 17 January 2022).